PERGAMON

Neural Networks 11 (1998) 623–635

Neural Networks

Contributed article

Assessing the importance of features for multi-layer perceptrons

Michael Egmont-Petersen[a,*], Jan L. Talmon[a], Arie Hasman[a], Anton W. Ambergen[b]

[a]Department of Medical Informatics, Maastricht University, Maastricht, The Netherlands
[b]Department of Methodology and Statistics, Maastricht University, Maastricht, The Netherlands

Received 28 May 1997; accepted 2 February 1998

Abstract

In this paper we establish a mathematical framework in which we develop measures for determining the contribution of individual features to the performance of a classifier. Corresponding to these measures, we design metrics that allow estimation of the importance of features for a specific multi-layer perceptron neural network. It is shown that all measures constitute lower bounds for the correctness that can be obtained when the feature under study is excluded and the classifier rebuilt. We also present a method for pruning input nodes from the network such that most of the knowledge encoded in its weights is retained. The proposed metrics and the pruning method are validated with a number of experiments with artificial classification tasks. The experiments indicate that the metric called replaceability results in the tightest error bounds. Both this metric and the metric called expected influence result in good rankings of the features. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords: Feature assessment; Feature selection; Neural networks; Bayes classifier; Pruning; Insight; Feature metrics; Feature measures

1. Introduction

Multi-layer perceptrons (MLPs) have been trained to perform various classification tasks (Cibas et al., 1996; Cunningham et al., 1992; Hansen et al., 1992; Harrison et al. (1991); Hart et al., 1989; Hripcsak, 1990; Moallemi, 1991; Poli et al., 1991; Schiøler et al., 1992; Schizas et al., 1990; Vogelsang, 1993). An MLP classifier performs a mapping from an input (feature or attribute) space onto an output (class) space. Cases are represented in the input space of the MLP by a vector that contains the *n* feature values of a case. The output vector of the MLP is used to classify a case, e.g. by means of the winner takes all rule. An MLP is an interesting alternative to other classifiers: even when the type of distribution of the features is unknown an MLP with the optimal number of hidden nodes approaches a Bayesian classifier and hence its error rate will be close to the minimum error rate (Richard et al., 1991).

For many classification tasks a large number of potentially useful features can be defined and added as input to the MLP. In these situations, feature selection is often a desired task. Ideally, when the acquisition costs of the features are equal, one wants to rank the available features according to the change in correctness that results from removing or adding the respective feature from the feature set (Siedlecki et al., 1988). We define the marginal contribution of a feature *k* among a set of *n* features as the difference in error rate of a classifier based on all *n* features and a classifier based on all but the *k*th feature.

Our goal is to estimate the marginal contribution of each feature used in a trained MLP and to prune the least important feature from the MLP without having to retrain its weights from scratch. First, we consider briefly different criteria for feature assessment and discuss approaches for feature selection that have been proposed in the literature. We then define four probabilistic measures that establish different upper bounds for the marginal contribution of a feature. These measures are made operational with metrics that make it possible to estimate these bounds for input features of a trained MLP. A method to prune features without having to train the MLP from scratch is also presented. The metrics and the pruning method are embedded in a backward search procedure and evaluated in a number of experiments.

2. Background

A number of approaches for feature selection have been developed (Batitti, 1994; Cibas et al., 1996; Foroutan et al.,

[*] Corresponding author. Division of Image Processing, Department of Radiology, Leiden University Medical Centre, P.O.B. 9600, NL-2300 RC Leiden, The Netherlands; E-mail: michael@lkeb.azl.nl.

0893-6080/98/$19.00 © 1998 Elsevier Science Ltd. All rights reserved.
PII S0893-6080(98)00031-8

1987; Holz et al., 1994; Karthaus et al., 1995; Kittler, 1980; Kudo et al., 1993; Siedlecki et al., 1988, 1989; Stahlberger et al., 1997). The best subset of features is obtained by a feature selection procedure. Such a procedure investigates different subsets of features according to a search scheme. At each step, the feature subsets are compared according to an assessment criterion. The procedure terminates when a satisfactory feature subset has been found.

2.1. Assessment criteria

Although the marginal contribution is the optimal criterion for assessing features, it is computationally complex to estimate the minimum error rate that can be obtained from a (sub)set of features in the general case where the type of distribution of the features is unknown. Alternative assessment criteria that are easier to compute have been suggested. Among these, probabilistic distance measures, dependence measures and entropy measures have been proposed (for overviews see Kittler, 1986; Siedlecki et al., 1988). With some distance measures, bounds of the error rate for the assessed feature subset can be determined. Most distance measures are inferior to the marginal contribution because their relationship with the error rate is often very loose (Kittler, 1986). Another drawback of using the probabilistic distance and dependency measures as assessment criteria is that they do not take into account the properties of a particular classifier, i.e. the contribution of each feature to classifier performance (Foroutan et al., 1987; Siedlecki et al., 1988).

A similar problem exists when a set of features that is optimal for one type of classifier is used as feature set for another type of classifier (Batitti, 1994). There is no guarantee that this feature set is also optimal for the other classifier.

2.2. Search schemes

A number of different search schemes exist. Besides an exhaustive search, which entails comparing $2^n - 1$ different subsets of features, suboptimal schemes such as forward and backward search as well as branch and bound search (Narenda et al., 1977) are the most frequently used. Examples of algorithms that select features by a forward search are NPPA (Talmon, 1986), ID3 (Quinlan, 1983), an approach to feature selection (Batitti, 1994) as well as a variant of stepwise discriminant analysis (Cooley et al., 1971). For applications of backward search see Nobis (1994) and Vogelsang (1993).

In practice, forward and backward search as well as the branch and bound scheme do not necessarily lead to the optimal subset of features. If the performance (on a test set) of an MLP is incidentally the best because of statistical fluctuations, one ends up exploring an inferior subset of features (Foroutan et al., 1987; Siedlecki et al., 1988). As a remedy, Siedlecky and Sklansky developed a genetic algorithm for feature selection (Siedlecki et al., 1989) and compared it with forward and backward search. Although the genetic algorithm outperformed both the forward, backward and branch and bound search schemes, their genetic approach is computationally much more complex. Their experiments showed that a backward search procedure yields ‘‘close to’’ optimal subsets of features (see also Foroutan et al., 1987).

Besides statistical fluctuations, backward search, when used for feature selection for MLPs, may lead to a suboptimal result because the standard learning algorithms do not guarantee convergence to the global minimum of the error function. When the MLP that is trained with the optimal subset of features ends up in a local minimum and its error rate exceeds the error rate of one of the other MLPs, the optimal subset of features is not explored further. In this case, backward search will result in a suboptimal set of features. The problem of local minima is usually remedied by training several MLPs, each with different initial weights and topologies. However, this is a lengthy procedure.

To overcome this problem we propose a new method for pruning an input node from a trained MLP. The pruning method adapts the weights that connect the other input nodes with the hidden nodes using the regression parameters which predict the feature that is to be pruned. Thereby, most of the knowledge embedded in the MLP is retained and retraining its weights from scratch may not be necessary. Recently, others have suggested the use of a variant of ‘‘optimal brain surgeon’’ (Hassibi et al., 1993) to prune input nodes (Cibas et al., 1996; Stahlberger et al., 1997). We propose a number of measures and metrics that can be used to guide the pruning process and to obtain estimates of the performance of the pruned MLP. The feature measures are all derived for a Bayesian classifier.

3. Four feature measures

We define a set of probability measures to estimate bounds of the marginal contribution of a feature to the performance of a statistical classifier. Each probability measure is made operational by a metric.

3.1. Classification

Classification is assigning a class label to a case based on an n-dimensional feature vector[1] $\boldsymbol{x}$. Let $\mathrm{p}(\boldsymbol{x}|\omega_j)$ denote the n-dimensional class-conditional probability density function (PDF) of the n features for class j, $j = 1,\ldots,c$. In general, classifiers partition the feature space into disjoint

[1] Henceforward, an uppercase letter X denotes a matrix, a bold letter $\boldsymbol{y}$ a column vector. $\boldsymbol{x}_i \in \mathrm{X}$ denotes column i in X, $\boldsymbol{x}^{\langle k \rangle}$ denotes row k in X, and $x_{k,i}$ the kth element in column i in X. The ith element in vector $\boldsymbol{y}$ is denoted by y_i. A function is in the main text rendered by f(•). In general, P(E) denotes the probability that the event E is observed, p(x) the probability density function of variable x.

regions R_j^n, $j = 1,\ldots,c$. For a minimum error-rate classifier, cases that occur in R_j^n have the highest posterior probability of belonging to class j and are classified as such. For such classifiers, R_j^n is given by (Anderson, 1958)

$$\mathrm{R}_j^n = \{\boldsymbol{x} \in \mathbb{R}^n | \mathrm{P}(\omega_j)\mathrm{p}(\boldsymbol{x}|\omega_j) > \mathrm{P}(\omega_l)\mathrm{p}(\boldsymbol{x}|\omega_l), \forall l \neq j\} \tag{1}$$

with $\mathrm{P}(\omega_j)$ the prior probability of class j. Denoting the probability of classifying a class j case correctly by $\mathrm{P}(\boldsymbol{x} \in \mathrm{R}_j^n|\omega_j)$, the correctness ρ^n of the classifier using n features becomes

$$\rho^n = \sum_{j=1}^{c} \mathrm{P}(\omega_j)\mathrm{P}(\boldsymbol{x} \in \mathrm{R}_j^n|\omega_j) \tag{2}$$

For a minimum error-rate classifier, the marginal contribution of a feature—the decrease in correctness that results when feature k is removed—is

$$\Delta\rho^{\neq k} = \sum_{j=1}^{c} \mathrm{P}(\omega_j)\big(\mathrm{P}(\boldsymbol{x} \in \mathrm{R}_j^n|\omega_j) - \mathrm{P}(\boldsymbol{x}^{\neq k} \in \mathrm{R}_j^{\neq k}|\omega_j)\big) \tag{3}$$

with $\boldsymbol{x}^{\neq k}$ an $n-1$ dimensional vector that is equal to $\boldsymbol{x}$ except for feature k that has been removed.

3.2. Feature measures

The probability $\mathrm{P}(\boldsymbol{x} \in \mathrm{R}_j^n|\omega_j)$ can be written as

$$\int_{\mathrm{R}_j^{n\backslash k}} \left[\int_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(\boldsymbol{x}|\omega_j)dx_k \right] d\boldsymbol{x}^{\neq k} \tag{4}$$

The range $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ is the set of x_k (for given $\boldsymbol{x}^{\neq k}$) for which $\boldsymbol{x}$ falls into R_j^n:

$$\mathrm{S}_j(\boldsymbol{x}^{\neq k}) = \{x_k \in \mathbb{R} | \mathrm{P}(\omega_j)\mathrm{p}(\boldsymbol{x}^{\neq k}, x_k|\omega_j) > \mathrm{P}(\omega_l)\mathrm{p}(\boldsymbol{x}^{\neq k}, x_k|\omega_l), \forall l \neq j\} \tag{5}$$

$\mathrm{R}_j^{n\backslash k}$ denotes the projection of the region R_j^n onto the $n-1$ dimensions excluding dimension k, i.e. $\mathrm{R}_j^{n\backslash k}$ is the set of $\boldsymbol{x}^{\neq k}$ for which $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ is not empty.

The larger the probability that x_k will fall in the range $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ the less feature k influences whether the case is classified into class j. Using the fact that $\mathrm{p}(\boldsymbol{x}^{\neq k},x_k|\omega_j) = \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{p}(x_k|\boldsymbol{x}^{\neq k},\omega_j)$, Eq. (5) can be written as

$$\mathrm{S}_j(\boldsymbol{x}^{\neq k}) = \left\{ x_k \in \mathbb{R} \left| \frac{\mathrm{p}(x_k|\boldsymbol{x}^{\neq k}, \omega_j)}{\mathrm{p}(x_k|\boldsymbol{x}^{\neq k}, \omega_l)} > \frac{\mathrm{P}(\omega_l)}{\mathrm{P}(\omega_j)} \frac{\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_l)}{\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)}, \forall l \neq j \right. \right\} \tag{6}$$

It is clear that $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ is determined by the relation between the feature-conditional likelihood ratio (left), the likelihood ratio of the $n-1$ other features (right) and the prior probabilities.

Eq. (4) can be rewritten as the integral over the product

$$\int_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j) \left[\int_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(x_k|\boldsymbol{x}^{\neq k}, \omega_j)\ dx_k \right] d\boldsymbol{x}^{\neq k} \tag{7}$$

and the correctness ρ^n as

$$\rho^n = \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j) \left[\int_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(x_k|\boldsymbol{x}^{\neq k}, \omega_j)\ dx_k \right] d\boldsymbol{x}^{\neq k} \tag{8}$$

Eq. (8) can be used to obtain some insight in the marginal contribution of feature k. We will approximate the integral over $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ in four different ways, on the basis of which we will estimate the contribution of feature k. In the following, we will discuss four different functions $\mathrm{g}_j(\boldsymbol{x}^{\neq k})$ that approximate the integral. We will implement these functions and study their usefulness with respect to selecting features to remove from trained neural networks. First, however, we prove that the reduction in correctness that is estimated with each of these four different functions $\mathrm{g}_j(\boldsymbol{x}^{\neq k})$, is always larger than or equal to the actual reduction that results when a feature is removed.

Theorem 3.2.1: The decrease in correctness $\Delta\rho^{\neq k}$ that results when feature k is removed is always smaller than or equal to $\rho^n - \rho'^{\neq k}$, thus $\Delta\rho^{\neq k} \leq \rho^n - \rho'^{\neq k}$.

Proof: By writing $\rho^n - \rho'^{\neq k}$ as

$$\rho^n - \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int_{\mathbb{R}^{n-1}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{g}_j(\boldsymbol{x}^{\neq k})\ d\boldsymbol{x}^{\neq k} \tag{9}$$

with g_j denoting c functions for which $0 \leq \mathrm{g}_j(\boldsymbol{x}^{\neq k}) \leq 1$, $\forall \boldsymbol{x}^{\neq k} \in \mathbb{R}^{n-1}$ and

$$\sum_{j=1}^{c} \mathrm{g}_j(\boldsymbol{x}^{\neq k}) \in \{0, 1\}, \forall \boldsymbol{x}^{\neq k} \in \mathbb{R}^{n-1} \tag{10}$$

the following can be derived

$$\sum_{j=1}^{c} \mathrm{P}(\omega_j) \int_{\mathbb{R}^{n-1}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{g}_j(\boldsymbol{x}^{\neq k})\ d\boldsymbol{x}^{\neq k} = \tag{11}$$

$$\int_{\mathbb{R}^{n-1}} \sum_{j=1}^{c} \mathrm{P}(\omega_j)\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{g}_j(\boldsymbol{x}^{\neq k})\ d\boldsymbol{x}^{\neq k} \leq \tag{12}$$

$$\int_{\mathbb{R}^{n-1}} \sum_{j=1}^{c} \{\max_{l=1,..,c}\big[\mathrm{P}(\omega_l)\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_l)\big]\}\mathrm{g}_j(\boldsymbol{x}^{\neq k})\ d\boldsymbol{x}^{\neq k} = \tag{13}$$

$$\int_{\mathbb{R}^{n-1}} \max_{l=1,\ldots,c}\big[\mathrm{P}(\omega_l)\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_l)\big] \sum_{j=1}^{c} \mathrm{g}_j(\boldsymbol{x}^{\neq k})\ d\boldsymbol{x}^{\neq k} \leq \tag{14}$$

$$\int_{\mathbb{R}^{n-1}} \max_{l=1,..,c}\big[\mathrm{P}(\omega_l)\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_l)\big] d\boldsymbol{x}^{\neq k} = \tag{15}$$

$$\int_{\mathbb{R}^{n-1}} \sum_{j=1}^{c} \mathrm{P}(\omega_j)\mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{I}(\boldsymbol{x}^{\neq k} \in \mathrm{R}_j^{\neq k})\ d\boldsymbol{x}^{\neq k} = \tag{16}$$

with I(.) denoting the indicator function that is 1 when $\boldsymbol{x}^{\neq k} \in \mathrm{R}_j^{\neq k}$, 0 otherwise. Eq. (16) is equal to

$$\sum_{j=1}^{c} \mathrm{P}(\omega_j) \int\limits_{\mathrm{R}_j^{\neq k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\; d\boldsymbol{x}^{\neq k} \tag{17}$$

with $\mathrm{R}_j^{\neq k}$, $j = 1,\ldots,c$ the regions of the optimal Bayes decision rule in $\mathbb{R}^{n-1}$. So $\rho^{\neq k} \geq \rho'^{\neq k}$ and therefore $\Delta\rho^{\neq k} \leq \rho^n - \rho'^{\neq k}$ holds.

By appropriately selecting the function $\mathrm{g}_j(\boldsymbol{x}^{\neq k})$ using information on the distribution of feature k, it is possible to minimize the gap between the marginal contribution of a feature $\Delta\rho^{\neq k}$ and the decrease in correctness $\rho^n - \rho'^{\neq k}$ that results from the substitution of feature k. We define four different measures that bound the maximum decrease in correctness that can occur when feature k is removed. These four different measures will be made operational in Section 4.

Let us assume that whenever feature k can influence the classification of the case, the case will be misclassified (that is whenever $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ is not equal to $\mathbb{R}$). This assumption implies that the integral

$$\int\limits_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(x_k|\boldsymbol{x}^{\neq k}, \omega_j)\; dx_k \tag{18}$$

in Eq. (8) is set equal to zero for $\mathrm{S}_j(\boldsymbol{x}^{\neq k}) \neq \mathbb{R}$. Therefore, we define the potential influence (ϕ_k) of feature k as

$$\phi_k \equiv \rho^n - \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int\limits_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{g}(\mathrm{S}_j(\boldsymbol{x}^{\neq k}) = \mathbb{R})\; d\boldsymbol{x}^{\neq k} \tag{19}$$

with

$$\mathrm{g}(e) = \begin{cases} 1: & e = \mathrm{TRUE} \\ 0: & e = \mathrm{FALSE} \end{cases} \tag{20}$$

Instead of the potential influence which clearly overestimates the contribution of feature k to ρ^n one can also try to estimate the contribution to ρ^n of that part of feature k that is independent of the other $n - 1$ features. The part of feature k that is dependent on the other $n - 1$ features is computed by its expected value given the values of the other features

$$\mathrm{E}(x_k|\boldsymbol{x}^{\neq k}) = \int_{-\infty}^{\infty} x_k \mathrm{p}(x_k|\boldsymbol{x}^{\neq k})\; dx_k \tag{21}$$

The difference between ρ^n and the resulting correctness, which we will call the replaceability (ι_k) of feature k, is an estimate of the contribution of the independent part of feature k

$$\iota_k \equiv \rho^n - \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int\limits_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j)\mathrm{g}\{\mathrm{E}(x_k|\boldsymbol{x}^{\neq k}) \in \mathrm{S}_j(\boldsymbol{x}^{\neq k})\} d\boldsymbol{x}^{\neq k} \tag{22}$$

The replaceability[2] is another and probably better estimate of the marginal contribution of feature k. In practice, $\mathrm{E}(x_k|\boldsymbol{x}^{\neq k})$ is unknown and will be substituted by an estimate of the population expected value denoted by $\hat{\mathrm{E}}(x_k|\boldsymbol{x}^{\neq k})$ as will become apparent in the operationalization of the measures in Section 4. We suggest a third measure that takes into account the stochastics of the model used to compute $\hat{\mathrm{E}}(x_k|\boldsymbol{x}^{\neq k})$ for a particular value of $\boldsymbol{x}^{\neq k}$. So we replace $\mathrm{g}\{\mathrm{E}(x_k|\boldsymbol{x}^{\neq k}) \in \mathrm{S}_j(\boldsymbol{x}^{\neq k})\}$ by a probability distribution $\mathrm{p}(\hat{\mathrm{E}}(x_k|\boldsymbol{x}^{\neq k})|\boldsymbol{x}^{\neq k})$. The difference between ρ^n and the resulting correctness we call the predicted influence (ζ_k) of feature k and is defined as

$$\zeta_k \equiv \rho^n - \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int\limits_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j) \times \int\limits_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(\hat{\mathrm{E}}(x_k|\boldsymbol{x}^{\neq k})|\boldsymbol{x}^{\neq k})\; dx_k\; d\boldsymbol{x}^{\neq k} \tag{23}$$

The potential influence was defined to identify poor features. However, as we have seen this measure overestimates the marginal contribution of feature k, because both the extent of $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ (when $\mathrm{S}_j(\boldsymbol{x}^{\neq k}) \neq \mathbb{R}$) and the probability of observing values of x_k in $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ are not taken into account: the more values observed in $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ the less the feature can influence the classification result. For a poor feature, moreover, one may expect that the difference between the marginal distribution $\mathrm{p}(x_k)$ and the conditional distribution $\mathrm{p}(x_k|\boldsymbol{x}^{\neq k},\omega_j)$ will be relatively small. If we therefore replace $\mathrm{p}(x_k|\boldsymbol{x}^{\neq k},\omega_j)$ by $\mathrm{p}(x_k)$ in Eq. (8), we obtain again an estimate of the influence of the feature. The difference in correctness between ρ^n and the resulting correctness we call the expected influence (ϱ_k) of feature k

$$\varrho_k \equiv \rho^n - \sum_{j=1}^{c} \mathrm{P}(\omega_j) \int\limits_{\mathrm{R}_j^{n\backslash k}} \mathrm{p}(\boldsymbol{x}^{\neq k}|\omega_j) \int\limits_{\mathrm{S}_j(\boldsymbol{x}^{\neq k})} \mathrm{p}(x_k)\; dx_k\; d\boldsymbol{x}^{\neq k} \tag{24}$$

Although $(\rho^n - \phi_k)$, $(\rho^n - \iota_k)$, $(\rho^n - \zeta_k)$, $(\rho^n - \varrho_k) \in (0,\rho^n)$, in practice one would never use a statistical classifier with a correctness smaller than

$$\sum_{j=1}^{c} \left(\mathrm{P}(\omega_j)\right)^2 \tag{25}$$

which is the correctness of a classifier that assigns the class labels at random, taking the prior distribution into account.

4. Four feature metrics

In practice, we estimate ϕ_k, ι_k, ζ_k and ϱ_k from a set of cases. For each case, the range $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ is obtained from a trained MLP. In Appendix A and Appendix B it is shown how this range can be found using a Taylor expansion. The numerical

[2] Note that a high replaceability is associated with a small value of ι_k and vice versa.

precision of the polynomial approximation is determined by the parameter $\varepsilon_{\max}$ which is derived in Appendix B.

4.1. Definition of an MLP

Let $\mathrm{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_r)$ denote a data matrix. A vector $\boldsymbol{x}_i$, which belongs to one of c classes, represents the n feature values of case i. Let α_k and β_k denote the lower and upper limits of feature k, respectively. α_k and β_k should be the minimum and maximum values that can possibly be observed for feature k.

Define a feed-forward MLP with one hidden layer with h hidden nodes as a mapping $\mathrm{N}:\{[\alpha_1,\beta_1],...,[\alpha_n,\beta_n]\} \rightarrow [\gamma,\eta]^c$:

$$\boldsymbol{o} = \mathrm{N}(\boldsymbol{x}) = \mathrm{f}(\mathrm{W}^2\mathrm{f}(\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1) - \boldsymbol{q}^2) \tag{26}$$

where W^1 is the weight matrix that connects the n input nodes with the h hidden nodes and W^2 the weight matrix connecting the h hidden with the c output nodes. $\boldsymbol{q}^1$ and $\boldsymbol{q}^2$ are the bias vectors of the hidden and output nodes, respectively. The function $\mathrm{f}: \mathbb{R}^{\dim(\boldsymbol{a})} \rightarrow [\gamma,\eta]^{\dim(\boldsymbol{a})}$ is the nonlinear, bounded activation function applied to each element in the activation vector $\boldsymbol{a}$, γ and η its lower an upper bound. Each element in the vector $\boldsymbol{o}$ represents the activation of a node in the output layer.

For MLPs that are used for classification tasks, each output node generally represents a class. Let the function class($\boldsymbol{o}$) denote the winner takes all rule which returns the index of the maximal element in the vector $\boldsymbol{o}$ or $\varnothing$ if two or more elements have the maximal value.

Define the matrix $\mathrm{E} = (\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_r)$, $\dim(\boldsymbol{e}_i) = c$, which specifies the correct class labels of the corresponding vectors in X. When $\boldsymbol{x}_i$ belongs to class j, $e_{j,i} = 1$ and $e_{l,i} = 0$, $\forall l \neq j$. The row vector $\boldsymbol{x}^{\langle k\rangle} = [x_{k,i}]_{i=1}^{r}$ contains the observations of feature k for all r cases.

4.2. Metrics

We first define a function change($\boldsymbol{x}$,$\boldsymbol{e}$,k) which for a given $\boldsymbol{x}^{\neq k}$ returns the set of values of $y = x_k$ that place $\boldsymbol{x}$ on the verge of misclassification with $\boldsymbol{e}$ indicating the correct class. Note that change (•) returns the empty set $\varnothing$ when $\forall y \in [\alpha_k,\beta_k]$: the case $\boldsymbol{x}$ always obtains the same class label or always an incorrect class label.

The potential influence of feature k is estimated from

$$\hat{\phi}_k = \hat{\rho}^n - \sum_{i=1}^{r} \frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times (1 - \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\})}{r} \tag{27}$$

which may be simplified to

$$\hat{\phi}_k = \sum_{i=1}^{r} \frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\}}{r} \tag{28}$$

the fraction of correctly classified cases for which feature k can influence the classification. The function card(S) returns the number of elements in the set S. The function $\mathrm{cor}(\boldsymbol{x},\boldsymbol{e}) = \mathrm{g}\{\mathrm{class}(\mathrm{N}(\boldsymbol{x})) = \mathrm{class}(\boldsymbol{e})\}$ is 1 when the vector $\boldsymbol{x}$ is classified correctly, otherwise 0. $\hat{\rho}^n$ is the estimated correctness using all n features. The metric $\hat{\phi}_k$ estimates ϕ_k using the MLP N($\boldsymbol{x}$).

Let us define the function z($\boldsymbol{x}$,$\boldsymbol{e}$,k) that returns the set of intervals S that together contain all values of $x_k \in [\alpha_k,\beta_k]$ which for given $\boldsymbol{x}^{\neq k}$ result in vector $\boldsymbol{x}$ being classified correctly. The ordered set $\mathrm{S} = \{\boldsymbol{s}(1), \boldsymbol{s}(2),\ldots, \boldsymbol{s}(t)\}, \boldsymbol{s}(d) = [\lambda_d, \upsilon_d]$, is an estimate of the range $\mathrm{S}_j(\boldsymbol{x}^{\neq k})$ defined in Eq. (5).

The replaceability of a feature is estimated by determining the decrease in correctness when feature k is substituted by its conditional mean. To estimate the replaceability of a feature, row k in X, $\boldsymbol{x}^{\langle k\rangle}$, is substituted with the values $\hat{\boldsymbol{x}}^{\langle k\rangle}$ predicted by multiple linear regression and the modified cases X′ are then classified with N.

Write the matrix $\mathrm{X} = [\mathrm{X}^{\langle k}: \boldsymbol{x}^{\langle k\rangle}:\mathrm{X}^{\rangle k}]^{\mathrm{T}}$ where $\mathrm{X}^{\langle k}$ denotes the submatrix composed of the (row) vectors $\boldsymbol{x}^{\langle 1\rangle},\ldots, \boldsymbol{x}^{\langle k-1\rangle}$ and $\mathrm{X}^{k\rangle}$ denotes the submatrix composed of the vectors $\boldsymbol{x}^{\langle k+1\rangle},\ldots, \boldsymbol{x}^{\langle n\rangle}$. Define the matrix $\mathrm{G}_k = [\mathrm{X}^{\langle k}:\boldsymbol{u}^{\mathrm{T}}:\mathrm{X}^{\rangle k}]^{\mathrm{T}}$, $\dim(\mathrm{G}_k) = n \times r$, $\dim(\boldsymbol{u}) = r$, $u_i = 1$, $i = 1,\ldots,r$. The predicted value $\hat{\boldsymbol{x}}^{\langle k\rangle}$ is computed as $\hat{\boldsymbol{x}}^{\langle k\rangle} = \boldsymbol{b}^{k\,\mathrm{T}}\mathrm{G}_k$ with $\boldsymbol{b}^k$ the least mean square (LMS) regression parameters estimated from the equation $\boldsymbol{b}^{k\ \mathrm{T}} = \boldsymbol{x}^{\langle k\rangle}\mathrm{G}_k^{\mathrm{T}}(\mathrm{G}_k\mathrm{G}_k^{\mathbf{T}})^{-1}$, $\dim(\boldsymbol{b}^k) = n$ and the element b_k^k the constant term. Using the function

$$\mathrm{N}^*(\boldsymbol{x}, k, y) = \mathrm{N}((x_1, x_2, \ldots, x_{k-1}, y, x_{k+1}, \ldots, x_n)^{\mathrm{T}}) \tag{29}$$

the replaceability ι_k of feature k is estimated from

$$\hat{\iota}_k = \hat{\rho}^n - \sum_{i=1}^{r} \frac{\mathrm{g}\{\mathrm{class}(\mathrm{N}^*(\boldsymbol{x}_i, k, \boldsymbol{b}^{k\,\mathrm{T}}\boldsymbol{g}_i^k)) = \mathrm{class}(\boldsymbol{e}_i)\}}{r} \tag{30}$$

$\hat{\iota}_k$ is the change in classifier correctness when feature k is replaced by its predicted value $\hat{\boldsymbol{x}}^{\langle k\rangle}$.

The parameters of the regression $\boldsymbol{b}^k$ used to compute the replaceability $\hat{\iota}_k$ have a stochastic component. For normally distributed features p($\boldsymbol{x}$), the predicted values estimated from $\hat{\boldsymbol{x}}^{\langle k\rangle} = \boldsymbol{b}^{k\,\mathrm{T}}\boldsymbol{g}_i^k$ are t-distributed around their true mean $\tilde{\boldsymbol{x}}^{\langle k\rangle} = \boldsymbol{\beta}^{k\,\mathrm{T}}\boldsymbol{g}_i^k$ with the variance $\hat{\mathrm{V}} = \hat{\sigma}^2_{res,k}\, \boldsymbol{g}^k(\mathrm{G}_k\ \mathrm{G}_k^{\mathrm{T}})^{-1}\boldsymbol{g}^{k\,\mathrm{T}}$ (Montgomery et al., 1992). $\hat{\sigma}^2_{res,k}$ is the residual variance of the regression estimated from

$$\hat{\sigma}^2_{res,k} = \frac{\|\, \hat{\boldsymbol{x}}^{\langle k\rangle} - \boldsymbol{x}^{\langle k\rangle}\|^2}{r - n} \tag{31}$$

and ||•|| the Euclidian vector norm.

The probability of observing $\hat{x}_{k,i}$ in the range S can be estimated as

$$\mathrm{pc}(\mathrm{S}, \hat{x}_k, r-n) = \sum_{s(d)\in \mathrm{S}} \mathrm{F}\left(\frac{\upsilon_d - \hat{x}_k}{\hat{\mathrm{V}}}, r-n\right) - \mathrm{F}\left(\frac{\lambda_d - \hat{x}_k}{\hat{\mathrm{V}}}, r-n\right) \tag{32}$$

with F the cumulative t-distribution (Parzen, 1960) with $r - n$ degrees of freedom.

The predicted influence $\hat{\zeta}_k$ of feature k can now be estimated from

$$\hat{\zeta}_k = \hat{\rho}^n - \sum_{i=1}^{r} \frac{\mathrm{pc}(\mathrm{z}(\boldsymbol{x}_i, \boldsymbol{e}_i, k), \boldsymbol{b}^{k\,\mathrm{T}} \boldsymbol{g}_i^k, r-n)}{r} \tag{33}$$

$\hat{\zeta}_k$ is the change in classifier correctness when feature k is replaced by its predicted value $\hat{\boldsymbol{x}}^{\langle k \rangle}$ taking the stochastics of the estimated regression vector $\boldsymbol{b}^k$ into account. This metric is an unbiased estimator of ζ_k when the n features are multivariate normally distributed, an assumption that also holds for $\hat{\iota}_k$. In practical situations this is hardly ever the case and the practical value of the metrics has to be established empirically.

The metric for the expected influence ϱ_k is computed from

$$\hat{\varrho}_k = \hat{\rho}^n - \sum_{i=1}^{r} \frac{\mathrm{pr}_k(\mathrm{z}(\boldsymbol{x}_i, \boldsymbol{e}_i, k))}{r} \tag{34}$$

The function $\mathrm{pr}_k(\mathrm{S})$ returns the probability that feature k is observed in one of the intervals in S

$$\mathrm{pr}_k(\mathrm{S}) = \sum_{\boldsymbol{s}(d) \in \mathrm{S}} \int_{\lambda_d}^{v_d} \mathrm{p}(x_k) dx \tag{35}$$

with $\mathrm{p}(x_k)$ the marginal PDF of feature k estimated from X. $\hat{\varrho}_k$ is an estimate of the change in classifier correctness ϱ_k when feature k is replaced by a value from the marginal distribution $\mathrm{p}(x_k)$.

4.3. Properties of the feature metrics

Two of the metrics, $\hat{\iota}_k$ and $\hat{\zeta}_k$, are based on the assumption that the features are normally distributed. Often this is not the case, so we briefly investigate the relation between these two metrics and the (nonparametric) metric $\hat{\phi}_k$.

Theorem 4.3.1: For an MLP, N, the potential influence is greater than or equal to the replaceability $\hat{\phi}_k \geq \hat{\iota}_k$.

Proof: The inequality $\hat{\phi}_k \geq \hat{\iota}_k$ may be written as

$$\hat{\rho} - \sum_{i=1}^{r} \frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times (1 - \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\})}{r} \geq \hat{\rho} - \sum_{i=1}^{r} \frac{\mathrm{g}\{\mathrm{class}(\mathrm{N}^*(\boldsymbol{x}_i, k, \boldsymbol{b}^{k\,\mathrm{T}} \boldsymbol{g}_i^k)) = \mathrm{class}(\mathrm{N}^*(\boldsymbol{e}_i))\}}{r} \tag{36}$$

It can be proved that for case i

$$\frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times (1 - \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\})}{r} \leq \frac{\mathrm{g}\{\mathrm{class}(\mathrm{N}^*(\boldsymbol{x}_i, k, \boldsymbol{b}^{k\,\mathrm{T}} \boldsymbol{g}_i^k)) = \mathrm{class}(\mathrm{N}^*(\boldsymbol{e}_i))\}}{r} \tag{37}$$

as the left-hand side in the inequality is 1 if and only if the case $\boldsymbol{x}_i$ is classified correctly and feature k has no influence on its classification. Consequently, feature $x_{k,i}$ can be replaced by any value $\hat{x}_{k,i} \in [\alpha_k, \beta_k]$. In this case the right-hand side is also 1. This second fraction can also be 1 when the left-hand side is 0. This happens when $\mathrm{S}_j(\boldsymbol{x}^{\neq k}) \neq \mathbb{R}$ and $x_{k,i} \in \mathrm{S}_j$ ($S_j \neq \emptyset$). Hence

$$\sum_{i=1}^{r} \frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times (1 - \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\})}{r} \leq \sum_{i=1}^{r} \frac{\mathrm{g}\{\mathrm{class}(\mathrm{N}^*(\boldsymbol{x}_i, k, \boldsymbol{b}^{k\,\mathrm{T}} \boldsymbol{g}_i^k)) = \mathrm{class}(\mathrm{N}^*(\boldsymbol{e}_i))\}}{r} \tag{38}$$

and $\hat{\phi}_k \geq \hat{\iota}_k$

Theorem 4.3.2: For an MLP, N, the potential influence is greater than or equal to the predicted influence $\hat{\phi}_k \geq \hat{\zeta}_k$ when $\alpha_k = -\infty$ and $\beta_k = \infty$.

Proof: This proof follows the same line as the proof of Theorem 4.3.1. For each case holds

$$\frac{\mathrm{cor}(\boldsymbol{x}_i, \boldsymbol{e}_i) \times (1 - \min\{\mathrm{card}(\mathrm{change}(\boldsymbol{x}_i, \boldsymbol{e}_i, k)), 1\})}{r} \leq \frac{\mathrm{pc}(\mathrm{z}(\boldsymbol{x}_i, \boldsymbol{e}_i, k), \boldsymbol{b}^{k\,\mathrm{T}} \boldsymbol{g}_i^k, r-n)}{r} \tag{39}$$

when $\alpha_k = -\infty$ and $\beta_k = \infty$. The assumption ensures that the right-hand side is 1 when the case $\boldsymbol{x}_i$ is classified correctly and feature k has no influence on its classification. Hence $\hat{\phi}_k \geq \hat{\zeta}_k$

5. Feature pruning

Each of the four feature metrics defined in the previous section estimates a bound for the marginal contribution of a feature. The feature metrics can be used as criteria to select features to be pruned from an MLP.

We developed a technique for pruning an input node from a trained MLP that in many situations makes retraining superfluous. Let us define LMS pruning:

Definition 5.1: Least mean square (LMS) pruning of a feature k from an MLP N consists of creating an MLP N′ identical to N but without input node k. The weights of N′ that connect the n − 1 input nodes with the h hidden nodes as well as their bias terms acquire values such that N′ classifies a set of cases identically as N does when feature k is replaced by $\hat{\boldsymbol{x}}^{\langle k \rangle}$, its LMS-predicted value.

LMS-pruning is obtained as follows. Assume for simplicity that input n is to be pruned from N. Define $\boldsymbol{a}$ as the input vector to the hidden nodes before the activation function f(•) is applied, see Eq. (26):

$$\boldsymbol{a} = \mathrm{W}^1 \boldsymbol{x} - \boldsymbol{q}^1 \tag{40}$$

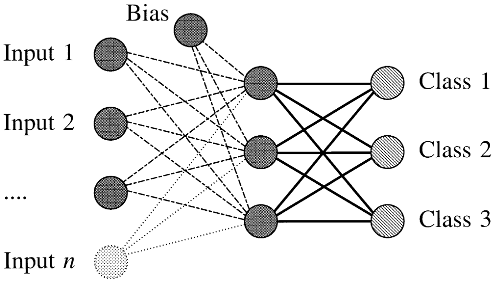

Fig. 1. Weights on the dotted connections are removed by LMS-pruning. The dashed weights are modified.

From the sample X, compute the regression parameter vector $\boldsymbol{b}^n$ and split it into the coefficient vector $\boldsymbol{b}^{nc}$ and the constant term b_n^n, $\boldsymbol{b}^n = (\boldsymbol{b}^{nc\ \mathrm{T}}, b_n^n)^{\mathrm{T}}$. We replace x_n in Eq. (40) by its predicted value $\hat{x}_{n,i}$ (for case i) computed from

$$\hat{x}_{n,i} = b_n^n + \sum_{j=1}^{n-1} b_j^{nc} x_{j,i} \tag{41}$$

Combining Eqs. (40) and (41) gives for hidden node u

$$a_u = \sum_{j=1}^{n-1} w_{u,j}^1 \, x_{j,i} - q_u^1 + w_{u,n}^1 \left(b_n^n + \sum_{j=1}^{n-1} b_j^{nc} x_{j,i} \right) \tag{42}$$

which simplifies to

$$a_u = \sum_{j=1}^{n-1} (w_{u,j}^1 + w_{u,n}^1 b_j^{nc}) x_{j,i} - (q_u^1 - w_{u,n}^1 b_n^n) \tag{43}$$

Define $\xi_{u,j}^1 = (w_{u,j}^1 + w_{u,n}^1 b_j^{nc})$, $\theta_u^1 = q_u^1 - w_{u,n}^1 b_n^n$ and construct a new MLP N′ that has $n - 1$ input nodes and the same number of hidden nodes as N with the weight matrix W^2, the bias vector $\boldsymbol{q}^2$, and the new weight matrix Ξ^1 and bias vector θ^1. Now feature $\boldsymbol{x}^{\langle n \rangle}$ has been LMS-pruned from N.

Fig. 1 illustrates which weights are modified (dashed) and which are pruned (dotted) when feature n is pruned. Replacing feature k with its conditional mean enforces a new partitioning of the class space. The boundaries that separate the new regions are determined by the intersection between the conditional mean (as function of the features $\boldsymbol{x}^{\neq k}$) and the class boundaries given by R$_j^n$, $j = 1,\ldots,c$.

The pruning operation turns out to be useful because the following holds.

Corollary 5.2: A network N with a correctness $\hat{\rho}^n$ *will, when feature k is LMS-pruned, have a correctness* $\rho' = \hat{\rho}^n - \hat{\iota}_k$.

This corollary specifies a lower bound for the correctness of a network from which feature k has been pruned as it is possible to retrain the LMS-pruned network using the weights of N′ as initial weight configuration and thereby possibly improve its correctness.

6. Experiments

We conducted a set of experiments to assess the developed metrics and the pruning method introduced in Section 5. We constructed two artificial classification problems to investigate whether the features were ranked correctly by each of the feature metrics. For each classification problem, the minimum error rate is computed analytically.

6.1. First experiment

In the first problem two classes A and B were characterized by 6 features with the centres $\mu_{\mathrm{A}} = (0,0,0,0,0,0)^{\mathrm{T}}$ and $\mu_{\mathrm{B}} = (1.75,\ 1.50,\ 1.25,\ 1.00,\ 0.75,\ 0.50)^{\mathrm{T}}$, respectively. Feature 1 has the largest discriminative power, feature 6 the smallest. We sampled 500 uncorrelated observations from the normal distribution D($\boldsymbol{x}|\mu_{\mathrm{A}}$,I) and 500 from D($\boldsymbol{x}|\mu_{\mathrm{B}}$,I) with I the identity matrix. The observations were divided into a training set and a test set each containing 250 vectors from class A and 250 from class B.

In total 30 MLPs with 2 hidden nodes, all with different initial weight configurations, were trained for 700 cycles with back-propagation in offline mode. The average correctness of the MLPs for the test set was $\rho_{avg} = 0.9274$ (± 0.0027). This correctness is very close to the Bayesian correctness, $\rho_{bayes} = 0.9292$.

We used Kendall's measure T$_{\mathrm{c}}$ for the correlation between several judges and a criterion ranking (Siegel et al., 1988) to compare the true (criterion) ranking of the 6 features with the ranking obtained from each feature metric. Table 1 shows the average rank order correlation T$_{\mathrm{c}}$ between the 30 MLPs and the true ranking that follows from the parameters μ_{A} and μ_{B}.

The first row in Table 1 shows that potential influence $\hat{\phi}_k$ is the poorest ranking criterion whereas the expected influence $\hat{\varrho}_k$ resulted in an optimal ranking ($\varepsilon_{\max} = 0.01$). The latter is to be expected as the features are independent (within the two classes). The predicted influence and the replaceability are slightly worse ranking criteria. An analysis of the weights of the MLPs indicated that feature 6 was given a larger weight than feature 5 in most of the 30 MLPs.

Table 1
The rank correlations T$_{\mathrm{c}}$ between the feature raking of the 30 MLPs and the true ranking. These are computed for two numerical precision levels $\varepsilon_{\max}$ of the polynomial approximation

	T$_{\mathrm{c}}$ (pot. infl.)	T$_{\mathrm{c}}$ (exp. infl.)	T$_{\mathrm{c}}$ (pred. infl.)	T$_{\mathrm{c}}$ (repl.)
$\varepsilon_{\max} = 0.01$	0.813	1.000	0.884	0.920
$\varepsilon_{\max} = 0.0001$	0.778	1.000	0.924	

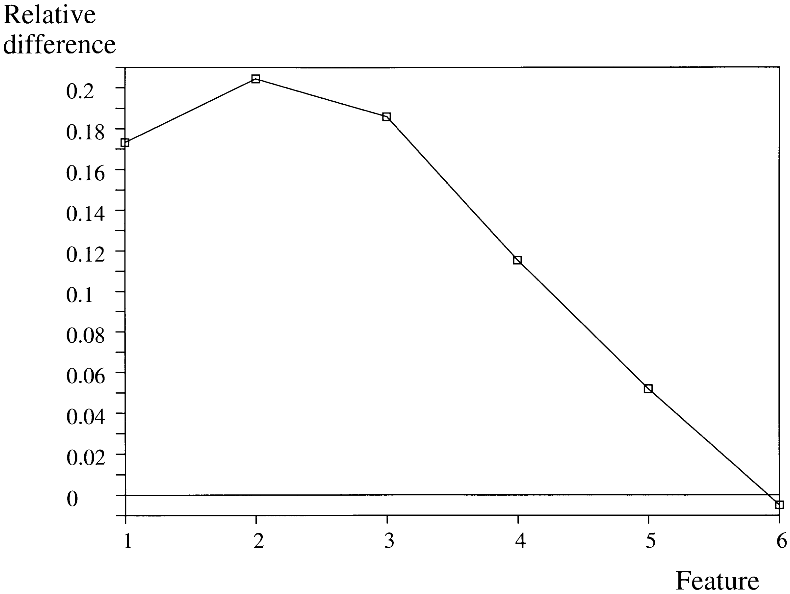

Fig. 2. Average relative difference between the potential influence as computed with the precision levels $\varepsilon_{\max} = 0.01$ and $\varepsilon_{\max} = 0.0001$.

6.2. Second experiment

In the second experiment, we investigated the influence of the numerical precision $\varepsilon_{\max}$ on the feature metrics. We recomputed all feature metrics except the replaceability ι_k (because ι_k does not depend on $\varepsilon_{\max}$) using the 30 MLPs from the first experiment with a higher precision level for the polynomial approximation, $\varepsilon_{\max} = 0.0001$. The second row in Table 1 shows the coefficient of agreement T_c between the true raking and the average ranking assigned by each metric to the features in the 30 MLPs with the increased precision level. The agreement between the predicted influence and the true rank slightly improves.

The feature metric that was influenced most by the level of precision is the potential influence. Fig. 2 shows the relative discrepancies between the potential influence computed for the six features, for both prediction levels of the polynomial approximation, $\varepsilon_{\max} = 0.01$ and $\varepsilon_{\max} = 0.0001$, $[\hat{\phi}_k(\varepsilon_{\max} = 0.01) - \hat{\phi}_k(\varepsilon_{\max} = 0.0001)]/\hat{\phi}_k(\varepsilon_{\max} = 0.01)$. Table 2 shows the potential influence of the six features.

The discrepancies between $\hat{\phi}_k(\varepsilon_{\max} = 0.01)$ and $\hat{\phi}_k(\varepsilon_{\max} = 0.0001)$ become small when the features are unimportant. This is also to be expected. For unimportant features, small fluctuations of the polynomial approximation around the true difference in output $o_j - o_l$ are unlikely to lead to false zero crossings, because $o_j - o_l$ is in more cases unequal to zero when feature x_k is varied within its range.

Table 2
Potential influence computed for the two different levels of precision for each of the six features

Precision	1	2	3	4	5	6
0.01	0.865	0.795	0.799	0.519	0.117	0.135
0.0001	0.727	0.647	0.663	0.462	0.111	0.134

We investigated the correlation between some of the feature metrics. The correlations in Table 3 indicate that the replaceability and the predicted influence metrics are closely related, which is also to be expected from their definition. Also the expected and the predicted influence are correlated. The potential influence is almost independent of the two other influence metrics.

6.3. Third experiment

A third experiment was designed to investigate how effective LMS-pruning is and to compare the ranking of each metric with the true ranking when the features contain dependencies. We designed a classification problem with three classes A, B and C that are characterized by six features. The centra of A and B were identical to the previous experiments and $\mu_C = -\mu_B$. The three classes have identical covariance matrices $\Sigma_A = \Sigma_B = \Sigma_C$, see Table 4.

We sampled 500 vectors from $D(\boldsymbol{x}|\mu_A,\Sigma)$, 500 from $D(\boldsymbol{x}|\mu_B,\Sigma)$ and 500 from $D(\boldsymbol{x}|\mu_C,\Sigma)$. These were divided into a

Table 3
Correlations between the four feature metrics computed among the 30 MLPs with the precision level used in the second experiment

Feature	Pot. vs. exp. influence	Exp. vs. pred. infl.	Pred. infl. vs. repl.	Pot. vs. pred. infl.
1	0.000	0.117	0.247	0.006
2	0.280	0.295	0.515	0.123
3	0.003	0.565	0.295	0.004
4	0.380	0.945	0.966	0.364
5	0.045	0.806	0.919	0.000
6	0.001	0.301	0.640	0.102
Avg.	0.118	0.505	0.597	0.100

Table 4
The covariance matrix used in the third experiment

1.0	0.4	0.0	0.0	0.0	0.0
	1.0	0.0	0.0	−0.3	0.0
		1.0	0.3	0.0	0.0
			1.0	−0.4	0.7
				1.0	0.0
					1.0

training and a test set each consisting of 750 vectors. Thirty MLPs with 2 hidden nodes, all with different initial weight configurations, were trained for 2000 cycles.

Whereas the correctness of the Bayesian classifier is 0.9124, the correctness of each MLP on the test set was 0.9093. That all 30 MLPs have the same correctness is due to the fact that these networks have exactly the number of degrees of freedom required for this classification task. In the previous experiment, the networks had also two hidden nodes although only one was necessary for that dichotomous classification task.

Table 5 contains the marginal contribution of each feature for a Bayesian classifier and the true ranking of the features. The feature with the smallest marginal contribution is the correct one to prune.

The four feature metrics were used to estimate the importance of each feature among the 30 MLPs using the set of training vectors. The most replaceable feature (smallest ι_k) was LMS-pruned and the importance of the five remaining features was estimated among the 30 LMS-pruned MLPs. This procedure was continued until only the two features 2 and 5 remained. The pruned MLPs were not retrained. Again, we used Kendall's measure T_c to compare the true rank of the features with the ranking obtained by each feature metric. The correlation coefficients are shown in Table 6. The correlation is not always 1 between the true and the observed feature ranking. When the metric ranked the least important feature correctly, this is indicated with ‘‘#’’. The symbol ‘‘*’’ indicates that some of the features obtain the same ranking (ties).

Table 6 indicates that the metrics for the predicted influence and the replaceability are superior to the other two metrics. Another observation is that the potential influence produces ties when the number of features is below 6. So when the classification relies on a few features, their contribution can only be assessed by taking the probability density function of the features into account. The expected influence only resulted in a good ranking when the number of features was reduced to 3. We conclude that in this experiment where the features contain dependencies, the predicted influence and replaceability are the best ranking criteria.

Fig. 3 shows the decrease in the average correctness among the 30 MLPs when features are LMS-pruned. The correctness is estimated with a test set that also contains 750 cases. For this classification problem, the pruning method is effective as the difference between the observed and theoretical correctness remains small, even when the pruned MLPs were not retrained.

7. Discussion

Four measures were defined to assess the importance of a feature for a classifier. The measures were made operational by metrics. One could ask whether all four are needed to assess the importance of features. In our experiments, the replaceability and the predicted influence are the best ranking criteria when the features contain dependencies and the expected influence the best criterion when the features are uncorrelated.

The potential influence metric can aid the construction of classifiers for sequential classification tasks. Quinlan distinguishes between sequential and parallel classification tasks (Quinlan, 1993). In parallel classification tasks all features are relevant for the classification of each case. In sequential classification tasks only a few of the available features determine the class label for a specific case. Whether a feature is relevant when classifying a specific case, depends on the value of one (or more) of the other features. When an MLP has been trained for a classification task, the potential influence metric can be used to identify features that are only (potentially) relevant for a small subset of cases. The least important of the n features can then be LMS-pruned. The procedure can be repeated for $n - 2$ features, etc. Thereby, the potential influence metric helps to establish the order in which features can be used by a sequential

Table 5
The true marginal contributions and feature rankings for a Bayesian classifier after successively removing the least contributing feature

Feature:	1	2	3	4	5	6
M. contribution	0.0125	0.0426	0.0011	0.0398	0.0643	0.0137
Ranking	5	2	6	3	1	4
M. contribution	0.0123	0.0473		0.0785	0.0910	0.0227
Ranking	5	3		2	1	4
M. contribution		0.1371		0.1073	0.1401	0.0302
Ranking		2		3	1	4
M. contribution		0.1504		0.0872	0.1137	
Ranking		1		3	2	
M. contribution		0.2534			0.0838	
Ranking		1			2	

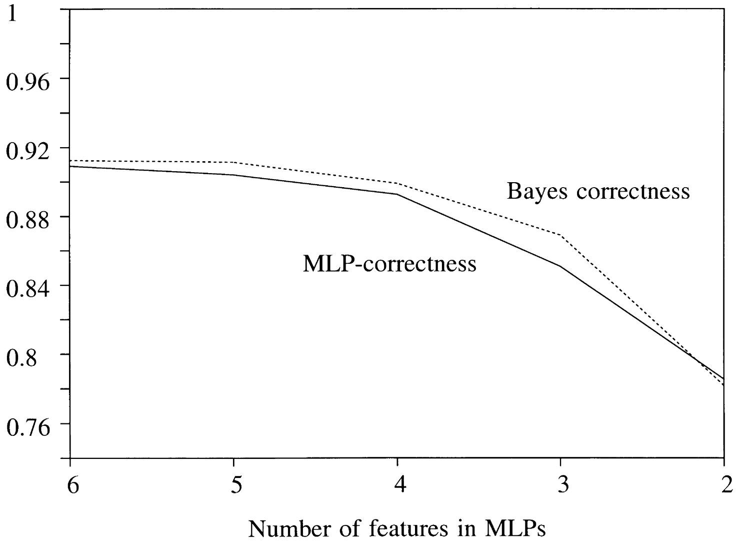

Fig. 3. Average decrease in correctness among the 30 MLPs lies close to the Bayes optimal correctness.

classifier, e.g. a cascade of MLPs. Building such a cascaded MLP classifier is, however, not trivial as the networks that are based on only a subset of features should be able to leave cases unclassified that can only be classified correctly using additional features.

The estimates computed with the four metrics all have a certain variance. In some cases, one might want to test whether the difference between two features with respect to a measure is significant or not. Consequently, one needs to know the underlying distribution of each estimate. We leave this issue for further research.

The major advantage of LMS-pruning is that one can prune a feature from a good MLP without having to train its weights from scratch. The amount of computation needed by a backward search is reduced as one does not need to train a set of networks with different initial weight configurations for each combination of $n - 1$ features. When a good subset of features has been identified, one can always try to retrain the MLP and possibly improve its performance. Our approach does not take into account that the number of hidden nodes that is optimal when using n features may not be optimal for $n - 1$ features. How to prune hidden nodes is left as a topic for further research.

Table 6
Correlation between the true ranking and the ranking obtained by each of the four feature metrics as a function of the number of input nodes of the MLPs

Features contained	6	5	4	3	2
T_c (pot. infl.)	0.867 #	0.738 *	0.707 *#	0.817 *#	0.000 *
T_c (exp. infl.)	0.733 #	0.600	0.000 #	1.000 #	1.000 #
T_c (pred. infl.)	0.867 #	1.000 #	1.000 #	1.000 #	1.000 #
T_c (repl.)	0.867 #	1.000 #	1.000 #	1.000 #	1.000 #

# Indicates that the least important feature was always correctly assessed.
* Indicates ties.

The overall correctness of a classifier is one of many possible yardsticks that can be used to measure the importance of a feature. If one wants an assessment that is independent of the prior probability of each class, the class-conditional correctness can be used as criterion (Egmont-Petersen et al., 1994). Class-conditional variants of our feature metrics can be easily computed by summing only over cases that belong to a given class.

We developed a numerical approach based on Taylor expansions to solve the $c - 1$ equations that determine the values of each feature for which two outputs of the MLP are equal. The polynomial approach solves the equations with sufficient accuracy but is computationally heavy as a different set of polynomial coefficients has to be computed for each feature k in each feature vector $\boldsymbol{x}_i$. Laguerre's method, which is used to find all roots in each polynomial, is also computationally complex. For one MLP with six input nodes, two hidden and three output nodes, the computation time for 750 vectors was 18 minutes on a Pentium-133 PC.

8. Conclusion

We defined a framework in which four measures for the importance of a feature for a classifier are developed. These measures are related to the marginal contribution of a feature. For each measure, we defined a metric to assess the importance of features for an MLP. It was suggested to use the metrics as ranking criteria to identify which features to prune from a trained MLP. When one wants to prune features according to a backward search scheme, we suggest the use of the replaceability as a raking criterion. This metric gives directly the correctness of the LMS-pruned MLP and the Taylor expansion is not needed to compute it.

Experiments illustrated that using LMS-pruning in combination with a backward search strategy enabled us to prune features from an MLP in an efficient way. The error rate obtained after a feature was LMS-pruned deviated only slightly from the Bayesian error rate. So in our experiments retraining the pruned networks from scratch could be avoided. We conclude that LMS-pruning is a convenient and computationally simple procedure to remove input nodes from an MLP.

Acknowledgements

The authors wish to thank the anonymous reviewers for valuable comments and suggestions.

Appendix A

In Section 4 we defined the function change(•). For a specific vector $\boldsymbol{x}$, this function returns values of feature k for which more than one element of the output $\boldsymbol{o} = \mathrm{N}(\boldsymbol{x})$ of the MLP N has the maximum value. To evaluate the function change(•), we need to identify the values of feature k in $[\alpha_k,\beta_k]$ that cause two output nodes to be maximal including the node that represents the correct class of the case. Given the vector $\boldsymbol{x}$, all values except x_k are kept fixed which allows us to write $o_j - o_l$ as a function of x_k. The roots of this equation comprise the values of feature k we seek. As all nodes different to j can be maximal, in total $c - 1$ equations need to be solved $o_j - o_l = 0$, $\forall l \neq j$. The subset of roots occurring in the interval $[\alpha_k,\beta_k]$ for which $o_j = \max(\boldsymbol{o})$, $j = \mathrm{class}(\boldsymbol{e})$, $\forall l \neq j$, constitutes the set of values to be returned by change(•).

As the output value o_j is computed from

$$o_j = \mathrm{f}\left(\boldsymbol{w}^{\langle j \rangle 2}\mathrm{f}(\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1) - q_j^2\right) \tag{A.1}$$

the $c - 1$ equations can be written as

$$\mathrm{f}\left(\boldsymbol{w}^{\langle j \rangle 2}\mathrm{f}(\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1) - q_j^2\right) - \mathrm{f}\left(\boldsymbol{w}^{\langle l \rangle 2}\mathrm{f}(\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1) - q_l^2\right) = 0 \tag{A.2}$$

for $l \neq j$. Solutions to these equations are called zero crossings. As f(•) is a monotonous transformation and $\mathrm{f}(0) = 0$, simplifies to

$$\left(\boldsymbol{w}^{\langle j \rangle 2} - \boldsymbol{w}^{\langle l \rangle 2}\right)\mathrm{f}(\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1) - (q_j^2 - q_l^2) = 0 \tag{A.3}$$

Now, the expression $\mathrm{W}^1\boldsymbol{x} - \boldsymbol{q}^1$ can for hidden node u be written as

$$w_{u,k}^1 x_k + v_u^k \tag{A.4}$$

with

$$v_u^k = \sum_{i \neq k} w_{u,i}^1 x_i - q_u^1 \tag{A.5}$$

Substituting Eq. (A.4) in Eq. (A.3) gives

$$\left(\boldsymbol{w}^{\langle j \rangle 2} - \boldsymbol{w}^{\langle l \rangle 2}\right)\mathrm{f}(\boldsymbol{w}_k^1 x_k + \boldsymbol{v}^k) - (q_j^2 - q_l^2) = 0 \tag{A.6}$$

or written as summation over the h hidden nodes

$$\sum_{u=1}^{h}\left(w_{j,u}^2 - w_{l,u}^2\right)\mathrm{f}(w_{u,k}^1 x_k + v_u^k) - (q_j^2 - q_l^2) = 0 \tag{A.7}$$

These equations cannot be solved analytically due to the nonlinear function f(•).

We use a polynomial approximation to the nonlinear function specified as $\mathrm{f}(x_k) = \tanh(w_{u,k}^1 x_k + v_u^k)$. Its Taylor expansion is given by

$$\Pi_u(x_k) = \sum_{n=0}^{\infty} \frac{(x_k - x_{0k})^n}{n!}\mathrm{f}^{(n)}(x_{0k}) \tag{A.8}$$

We incorporate the constant x_{0k} into the coefficients of the polynomial using the binomial theorem

$$(a - b)^n = \sum_{t=0}^{n}\binom{n}{t}a^t(-b)^{n-t} \tag{A.9}$$

The tth coefficient (coefficient of $(x_k)^t$) of the polynomial $\Pi_u(x_k)$ becomes

$$\psi_{u,t} = \sum_{n=t}^{\infty} \frac{\binom{n}{t}(-x_{0k})^{n-t}}{n!}\mathrm{f}^{(n)}(x_{0k}) \tag{A.10}$$

The coefficients of the polynomial expansions are summed over the hidden nodes to obtain one polynomial that approximates the $c - 1$ equations $o_j - o_l = 0$, $l \neq k$

$$\sum_{t=0}^{\infty}\left(\sum_{u=1}^{h}(w_{j,u}^2 - w_{l,u}^2)\psi_{u,t}\right)(x_k)^t - (q_j^2 - q_l^2) = 0 \tag{A.11}$$

In practice, the degree of the polynomial expansion has to be limited. We have set the maximum degree to 4 and approximate $o_j - o_l$ with a number of concatenated polynomials. Each polynomial approximates $o_j - o_l$ with a specified precision in a subinterval $[x_{beg},x_{end}]$ of $[\alpha_k,\beta_k]$. Together, the polynomials provide an approximation over the whole interval. In Appendix B it is shown how the values x_{beg}, x_{end} and $\mathrm{f}^{(n)}(x_{0k})$ are computed. For a discussion of polynomial approximation of MLPs see Williamson et al. (1995).

We use Laguerre's method (Press et al., 1988) to find all real and complex roots of the polynomials. We discard complex roots and roots outside the interval in which each polynomial provides a sufficiently accurate approximation. A root is considered as complex when the value of its imaginary component exceeds the numerical precision $\varepsilon_{\max}$.

Appendix B

We introduce the approximation precision $\varepsilon_{\max} > 0$. The function $\mathrm{f}(w_{u,k}^1 x_k + v_u^k)$ for hidden node u is approximated

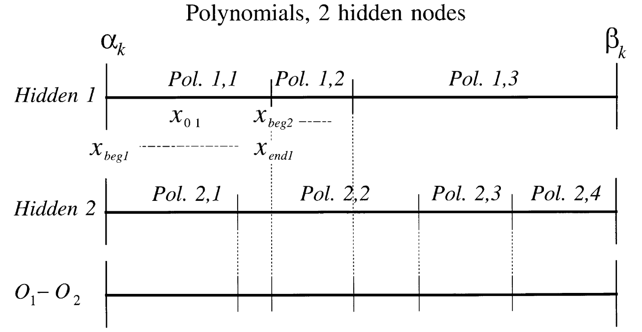

Fig. 4. Coefficients of the consecutive polynomials are chosen such that $\forall x \in [\alpha_k,\beta_k]$: $\varepsilon \leq \varepsilon_{\max}$ (see hidden node 1). The coefficients of the polynomials corresponding to each hidden node are added resulting in one polynomial pertaining to each intersecting interval.

with a series of Taylor polynomials, each with a different value of x_0. These values of x_0 are chosen such that the approximation interval $[x_{beg},x_{end}]$ of the respective polynomials together span the interval $[\alpha_k,\beta_k]$, see Fig. 4.

We use Lagrange's remainder formula to determine the approximation interval of each polynomial $[x_{beg},x_{end}]$ with $x_0 \in [x_{beg},x_{end}]$ for a given maximal approximative error $\varepsilon_{\max}$ (Ralston, 1965; Sydsæter, 1993):

$$\varepsilon_{\max} \leq \frac{|x-x_0|^{n+1}}{(n+1)!}M \tag{B.1}$$

where M is the maximum absolute value of $f^{(n+1)}(x)$, the $(n+1)$th derivative of $f(x)$, $\forall x \in m$ (for simplicity $m = \mathbb{R}$). For a fixed x_{beg} ($=\alpha_k$ for the first polynomial), when $\varepsilon_{\max}$ is specified, we may determine x_0 and x_{end} of the polynomial that guarantees an error smaller than $\varepsilon_{\max}$ by rearranging Eq. (B.1). Now solving for x_0 gives

$$\left(\frac{\varepsilon_{\max}(n+1)!}{M}\right)^{(n+1)^{-1}} \geq |x-x_0| \tag{B.2}$$

$$x_{beg} - \left(\frac{\varepsilon_{\max}(n+1)!}{M}\right)^{(n+1)^{-1}} = x_0 \tag{B.3}$$

and for x_{end}

$$x_0 + \left(\frac{\varepsilon_{\max}(n+1)!}{M}\right)^{(n+1)^{-1}} = x_{end} \tag{B.4}$$

The extreme value M can be found by solving $f^{(n+2)}(x) = 0$ and choosing the root that maximizes $|f^{(n+1)}(x)|$. The nth derivative $f^{(n)}(x)$ is defined as

$$f^{(n)}(x_0) = \frac{d^n}{dx^n}\tanh(x_0) \tag{B.5}$$

with x_0 a value in the domain of $\tanh(x)$.

We limit the number of the coefficients in a polynomial to 5. This allows us to use the roots of $f^{(6)}(x) = 0$ to determine the begin and end points of a polynomial $\Pi_u(x_k)$ with the degree 4. The fifth and sixth derivatives of the function $f(x) = \tanh(wx + v)$ with respect to x are

$$f^{(5)}(x) = 8w^5\,\frac{2\cosh(wx+v)^4 - 15\cosh(wx+v)^2 + 15}{\cosh(wx+v)^6} \tag{B.6}$$

$$f^{(6)}(x) = -16w^6\sinh(wx+v) \times \frac{2\cosh(wx+v)^4 - 30\cosh(wx+v)^2 + 45}{\cosh(wx+v)^7} \tag{B.7}$$

The roots $f^{(6)}(x) = 0$ are

$$-\frac{v}{w} \tag{B.8}$$

$$\pm\frac{\ln\left(\frac{1}{2}\sqrt{3}\sqrt{5+\sqrt{15}}\sqrt{2}+\sqrt{\frac{13}{2}+\frac{3}{2}\sqrt{15}}\right)-v}{w} \tag{B.9}$$

and

$$\pm\frac{\ln\left(\frac{1}{2}\sqrt{3}\sqrt{5-\sqrt{15}}\sqrt{2}+\sqrt{\frac{13}{2}-\frac{3}{2}\sqrt{15}}\right)-v}{w} \tag{B.10}$$

For each hidden node, the origin x_{0k} of the first polynomial is computed from Eq. (B.3) where the begin point $x_{beg1} = \alpha_k$, the smallest value feature k can possibly take. Then x_{end1} is computed from Eq. (B.4). The point x_{beg2} of the second polynomial is set equal to x_{end1}. This procedure is continued until x_{end} of a polynomial exceeds the limit β_k.

For an MLP with a number of hidden nodes, the polynomials specified in Eq. (A.11) have to be added taking into account the approximation interval of each polynomial. So, for example, for an MLP with 2 hidden nodes, the polynomial $\Pi_{1,1}$ approximates hidden node 1 in the interval $[x_{beg1,1},x_{end1,1}]$ and $\Pi_{2,1}$ hidden node 2 in the interval $[x_{beg2,1},x_{end2,1}]$, see Fig. 4. Now, we construct a polynomial that approximates $o_j - o_l$ by adding the coefficients of the two polynomials $\Pi_{1,1}$ and $\Pi_{2,1}$ of which the approximation intervals $[x_{beg1,1},x_{end1,1}]$ and $[x_{beg2,1},x_{end2,1}]$ overlap.

References

Anderson, T. W. (1958). *An introduction to multivariate statistical analysis*. New York: John Wiley.

Batitti R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks, 5 (4)*, 537–550.

Cibas T., Fogelman Soulié F., Gallinari P., & Raudys S. (1996). Variable selection with neural networks. *Neurocomputing*, *12*, 223–248.

Cooley, W. W., & Lohnes, P. R. (1971). *Multivariate data analysis*. New York: John Wiley.

Cunningham, J., & Haykin, S. (1992). Neural network detection of small moving radar targets in an ocean environment. In Kung, S. Y., Fallside, F., Sorenson, J. A., & Kaufmann, C. A. (Eds.), *Proceedings of the 1992 IEEE workshop on neural networks for signal processing* (pp. 306–315). NJ: IEEE.

Egmont-Petersen M., Talmon J. L., Brender J., & NcNair P. (1994). On the quality of neural net classifiers. *Artificial Intelligence in Medicine, 6 (5)*, 359–381.

Foroutan I., & Sklansky J. (1987). Feature selection for automatic classification of non-Gaussian data. *IEEE Transactions on Systems, Man, and Cybernetics, 17 (2)*, 187–198.

Hansen, L. K., Liisberg, C., & Salamon, P. (1992). Ensemble methods for handwritten digit recognition. In S. Y. Kung, F. Fallside, J. A. Sorenson, & C. A. Kaufmann (Eds.), *Proceedings of the 1992 IEEE workshop on neural networks for signal processing* (pp. 333–342). NJ: IEEE.

Harrison, R. F., Marshall, S. J., & Kennedy, R. L. (1991). A connectionist aid to the early diagnosis of myocardial infarction. In M. Stefanelli, A. Hasman, M. Fieschi, & J. Talmon (Eds.), *AIME-91*, Lecture Notes in Medical Informatics 44 (pp. 119–128). Berlin: Springer Verlag.

Hart, A., & Wyatt, J. (1989). Connectionist models in medicine: an investigation of their potential. In J. Hunter, J. Cookson, & J. Wyatt (Eds.), *AIME-89*, Lecture Notes in Medical Informatics 38 (pp. 115–124). Berlin: Springer Verlag.

Hassibi, B., & Stork, D. G. (1993). Second order derivatives for network pruning: optimal brain surgeon. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in Neural Information Processing Systems*, (pp. 164–171). MA: Morgan Kaufmann.

Holz, H. J., & Loew, M. H. (1994). Relative feature importance: a classifier-independent approach to feature selection. In E. S. Gelsema, & L. N. Kanal (Eds.), *Proceedings of pattern recognition in practice IV: Multiple paradigms, comparative studies and hybrid systems* (pp. 473–487). Amsterdam: Elsevier.

Hripcsak G. (1990). Using connectionistic modules for decision support. *Methods of Information in Medicine*, *29*, 167–181.

Karthaus V., Thygesen H., Egmont-Petersen M., Talmon J., Brender J., & McNair P. (1995). User-requirements driven learning. *Computer Methods and Programs in Biomedicine*, *48 (1-2)*, 39–44.

Kittler, J. (1980). Computational problems of feature selection pertaining to large data sets. In E. S. Gelsema, & L. N. Kanal (Eds.), *Proceedings of pattern recognition in practice* (pp. 405–414). Amsterdam: Elsevier.

Kittler, J. (1986). Feature selection and extraction. In T. Y. Young & K.-S. Fu (Eds.), *Handbook of Pattern Recognition and Image Processing*. Orlando: Academic Press.

Kudo M., & Shimbo M. (1993). Feature selection based on the structural indices of categories. *Pattern Recognition*, *26 (6)*, 891–901.

Moallemi C. (1991). Classifying cells for cancer diagnosis using neural networks. *IEEE Expert*, *12*, 8–12.

Montgomery, D. C., & Peck, E. A. (1992). *Introduction to Regression Analysis*, 2nd ed. New York: John Wiley.

Narenda P. M., & Fukunaga K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, *C-26 (9)*, 917–922.

Nobis, T. (1994). Berücksichtigung lokaler und globaler Textureigenschaften durch Erweiterung des Konzepts der Grauwertübergangsmatrizen auf einen Multi skalen ansatz, Diplomarbeit (Master Thesis), Aachen: Institut für Medizinische Informatik und Biometrie, Medizinische Fakultät, RWTH-Aachen.

Parzen, E. (1960). *Modern Probability Theory and its Applications*. New York: John Wiley.

Poli R., Cagnoni S., Livi R., Coppini G., & Valli G. (1991). A neural network expert system for diagnosing and treating hypertension. *IEEE Computer*, *3*, 64–71.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, V. T. (1988). *Numerical Recipes in C*. New York: Cambridge University Press.

Quinlan, J. R. (1983). Learning from noisy data. In *Proceedings of the third international machine learning workshop* (pp. 58–64).

Quinlan, J. R. (1993). Comparing connectionist and symbolic learning methods. In S. Hanson, G. Drastal, & R. Rivest (Eds.), *Computational Learning Theory and Natural Learning Systems: Constraints and Prospects*. Cambridge: MIT Press.

Ralston, A. (1965). *A First Course in Numerical Analysis*. Tokyo: McGraw-Hill.

Richard M. D., & Lippmann R. P. (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, *3*, 461–483.

Schiøler, T., Grimson, W., Sharpe, P., Egmont-Petersen, M., Momsen, G., O'Moore, R., & McNair, P. (1992). Automatic decision support based on voting by independent decision support systems. In *Proceedings of computing in clinical laboratories '92* (p. 58–66).

Schizas C. N., Pattchis C. S., Schofield I. S., & Fawcett P. R. (1990). Artificial neural nets in computer-aided macro motor unit potential classification. *Trans. IEEE Engineering in Medicine and Biology*, *9 (5)*, 31–38.

Siedlecki W., & Sklansky J. (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, *2 (2)*, 197–220.

Siedlecki W., & Sklansky J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, *10 (5)*, 335–347.

Siegel, S., & Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*, 2nd ed. Singapore: McGraw Hill.

Stahlberger, A., & Riedmiller, M. (1997). Fast network pruning and feature extraction using the unit-OBS algorithm. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, (pp. 655–660). MA: Morgan Kaufmann.

Sydsæter, K. (1993). *Matematisk Analyse. Bind 1*, 5th ed. Oslo: Universitetsforlaget.

Talmon J. L. (1986). A multiclass nonparametric partitioning algorithm. *Pattern Recognition Letters*, *4*, 31–38.

Vogelsang, F. (1993). Segmentierung radiologisch dokumentierter fokaler Knochenläsionen auf Basis kontextbezogener Vektoren mit neuronalen Netzwerken, Diplomarbeit (Master Thesis). Aachen: Institut für Medizinische Informatik und Biometrie, Medizinische Fakultät, RWTH-Aachen.

Williamson R. C., & Helmke U. (1995). Existence and uniqueness results for neural network approximations. *IEEE Trans. on Neural Networks*, *6 (1)*, 2–13.